NPS62-87-013

NAVAL POSTGRADUATE SCHOOL

Monterey, California

STUDY ON IMAGE PROCESSING
FOR
TURBID WATER VIEWING

Jae S. Lim

and

Charles W. Therrien

Final Report for Period April 20 - May 29, 1987

Prepared for:

Naval Undersea Warfare Engineering Station
Keyport, WA 98345

NAVAL POSTGRADUATE SCHOOL
Monterey, California

The work reported herein was supported in part by The Naval Undersea Warfare Engineering Station, Keyport, Washington.

This report was prepared by;

REPORT DOCUMENTATION PAGE

EPORT SECURITY CLASSIFICATION JNCLASSIFIED	1b RESTRICTIVE MARKINGS
ECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Unlimited
ECLASSIFICATION / DOWNGRADING SCHEDULE	
RFORMING ORGANIZATION REPORT NUMBER(S) PS62-87-013	5 MONITORING ORGANIZATION REPORT NUMBER(S)
AME OF PERFORMING ORGANIZATION aval Postgraduate School onterey, CA 93943-5000 — 6b OFFICE SYMBOL (If applicable) 62	7a NAME OF MONITORING ORGANIZATION
DDRESS (City, State, and ZIP Code) onterey, California 93943-5000	7b. ADDRESS (City, State, and ZIP Code)
ME OF FUNDING/SPONSORING GANIZATION UWES — 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N0025387WR70001
DDRESS (City, State, and ZIP Code) eyport, WA	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO. / WORK UNIT ACCESSION NO

LE (Include Security Classification)

tudy on Image Processing for Turbid Water Viewing

RSONAL AUTHOR(S)
ae S. Lim and Charles W. Therrien

PE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
	FROM 4/20/87 TO 5/29/87	Jun 1 1987	23

PLEMENTARY NOTATION

COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
LD	GROUP	SUB-GROUP	

TRACT (Continue on reverse if necessary and identify by block number)
1 effort to process images in turbid water viewing environment for possible improvement 1 image quality and intelligibility is in progress at the Naval Postgraduate School 1der the direction of Professor Charles Therrien. As a part of this effort we investi-ated the performance and computational requirements of an existing algorithm. We also eveloped new methods, investigated their computational requirements, and studied their xpected performance. Finally, we performed some very preliminary study on issues related the possible real time implementation of these algorithms. In this report, we describe e results of these efforts.

RIBUTION/AVAILABILITY OF ABSTRACT ICLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
ME OF RESPONSIBLE INDIVIDUAL arles Therrien	22b TELEPHONE (Include Area Code) ext. 3347	22c OFFICE SYMBOL 62Ti

TABLE OF CONTENTS

1 Introduction 2
2 Adaptive Filtering 3

2.1 Basic Algorithm 3
2.2 Parameter Choice and Expected Performance 5
2.3 Computational Requirements 5

3 Modified Adaptive Filtering 10

3.1 Basic Algorithm 10
3.2 Parameter Choice and Expected Performance 12
3.3 Computational Requirements 12

4 Temporal Filtering 16

4.1 Basic Algorithm 16
4.2 Parameter Choice and Expected Performance 17
4.3 Computational Requirements 17

5 Pseudo Color Representation 19
6 Hardware Implementation 20
7 References 23

1 Introduction

The recovery of torpedoes on the various NUWES test ranges is an important problem to the U.S. Navy. The recovery operation is carried on undersea with the aid of digging equipment and underwater video camera equipped with artificial lights. The digging operation is controlled and monitored on video monitors at the surface on board the recovery vessel.

The recovery of torpedoes is not a simple problem. Torpedoes are sometimes buried in the sediment and are difficult to locate. In addition, the recovery equipment in its attempt to dig out the torpedo stirs up sediment which visually obscures the object of interest and impedes the recovery operation. Since the human operator carrying on the recovery operation relies heavily on the video image received from the underwater camera and displayed on the monitor, improving the quality and intelligibility of the video image in this turbid water viewing environment has the potential to significantly increase the efficiency of the difficult recovery operation.

An effort to process images in turbid water viewing environment for possible improvement in image quality and intelligibility is in progress at the Naval Postgraduate School under the direction of Professor Charles Therrien. As a part of this effort, over the past six weeks we investigated the performance and computational requirements of an existing algorithm. We also developed new methods, investigated their computational requirements, and studied their expected performance. Finally, we performed some very preliminary study on issues related to the possible real time implementation of these algorithms. In this report, we describe the results of these efforts.

2 Adaptive Filtering

2.1 Basic Algorithm

Observation of video images in a turbid water environment shows that the contrast of images is often reduced significantly for a variety of reasons including the presence of sediment that is stirred in a digging operation. One approach that was previously considered at the Naval Postgraduate School and which we explored further as a part of this effort is to apply a contrast enhancement algorithm. The specific algorithm used is shown in Figure 1.

In the figure, $f(n_1, n_2)$ is a frame ($N \times N$ pixels) of the video data, $f_L(n_1, n_2)$ is the local luminance mean which is obtained by low-pass filtering $f(n_1, n_2)$, and $f_H(n_1, n_2)$ is the local contrast obtained by subtracting $f_L(n_1, n_2)$ from $f(n_1, n_2)$. The low-pass filtering operation is performed by local averaging over an $M \times M$ pixel region. The processed local contrast $\hat{f}_H(n_1, n_2)$ is obtained by multiplying $f_H(n_1, n_2)$ by the contrast enhancement factor $k(f_L)$. The processed local luminance mean $\hat{f}_L(n_1, n_2)$ is obtained by applying a point non-linearity function to $f_L(n_1, n_2)$. The processed image $p(n_1, n_2)$ is obtained by combining $\hat{f}_H(n_1, n_2)$ and $\hat{f}_L(n_1, n_2)$.

The algorithm in Figure 1 is capable of modifying the local contrast as a function of the local luminance mean and modifying the local luminance mean. The algorithm has been [1] successfully applied previously to improving the contrast of optical images taken from an airplane through varying amounts of cloud cover. Since degradations such as those due to the sediment stirred up during the recovery operation appear to reduce the image contrast in a manner similar to the contrast reduction due to cloud cover, this particular algorithm was chosen originally for a more detailed study.

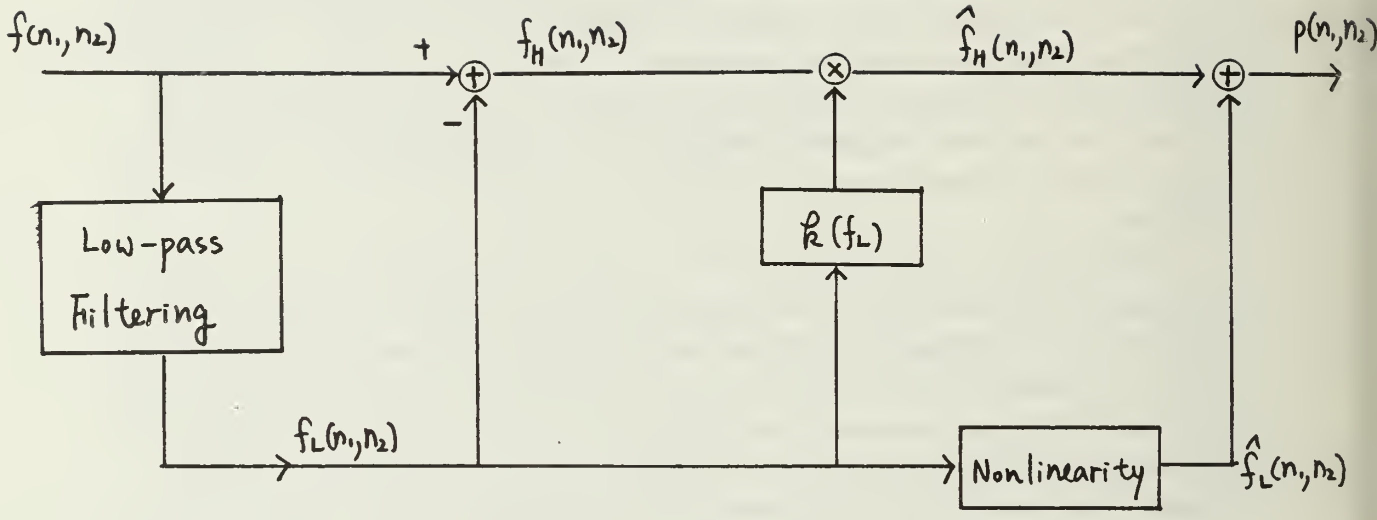

Figure 1. Adaptive Filtering Algorithm

2.2 Parameter Choice and Expected Performance

The algorithm in Figure 1 has been applied to a limited set of data from NUWES. From this study, we have observed that the data generally have a certain level of background noise and this limits the utility of the adaptive filtering algorithm. Any contrast enhancement method including the adaptive filtering method we have studied tends to emphasize high frequency components and as a result tends to emphasize the background noise. For the video data from NUWES, we have observed that the background noise becomes very visible when the contrast enhancement factor $k(f_L)$ is chosen above 3. This is in sharp contrast with the cloud cover data to which this algorithm was originally applied. The noise level in the cloud cover data was sufficiently low so that the contrast enhancement factor in the range of $6 \sim 8$ could be used without any visible noise in the processed image.

Based on the above considerations, the choice of $k(f_L)$ and the nonlinearity function we recommend is shown in Figures 2 and 3. The function $k(f_L)$ lies between 1.0 and 3.0, and increases as f_L increases. This feature exploits the notion that the same level of noise is more visible in the dark regions relative to the bright regions. We also recommend a mask size of 5×5 pixels for the low-pass filtering operation.

Since the contrast enhancement factgor $k(f_L)$ recommended is less than 3, a very large amount of contrast enhancement is not possible. However, the contrast enhancement by a factor of $2 \sim 3$ is still significant and we expect that noticeable contrast enhancement is still possible with the recommended choice of parameters.

2.3 Computational Requirements

All our discussions on computational requirements in this and future sections are based on the assumption that $f(n_1, n_2)$ is an image frame of $N \times N$ pixels, the low-pass filtering operation is a simple average over a region of $M \times M$ pixels, and R frames are processed per second.

The major computations involved in processing one image frame using the adaptive filtering algorithm are as follows:

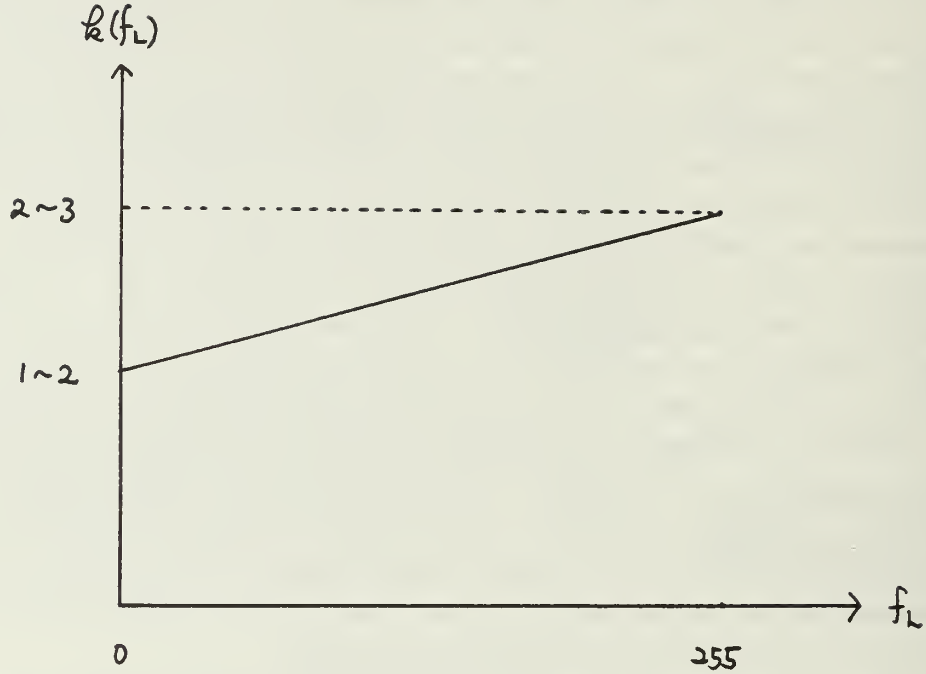

Figure 2. Recommended Choice of $k(f_L)$

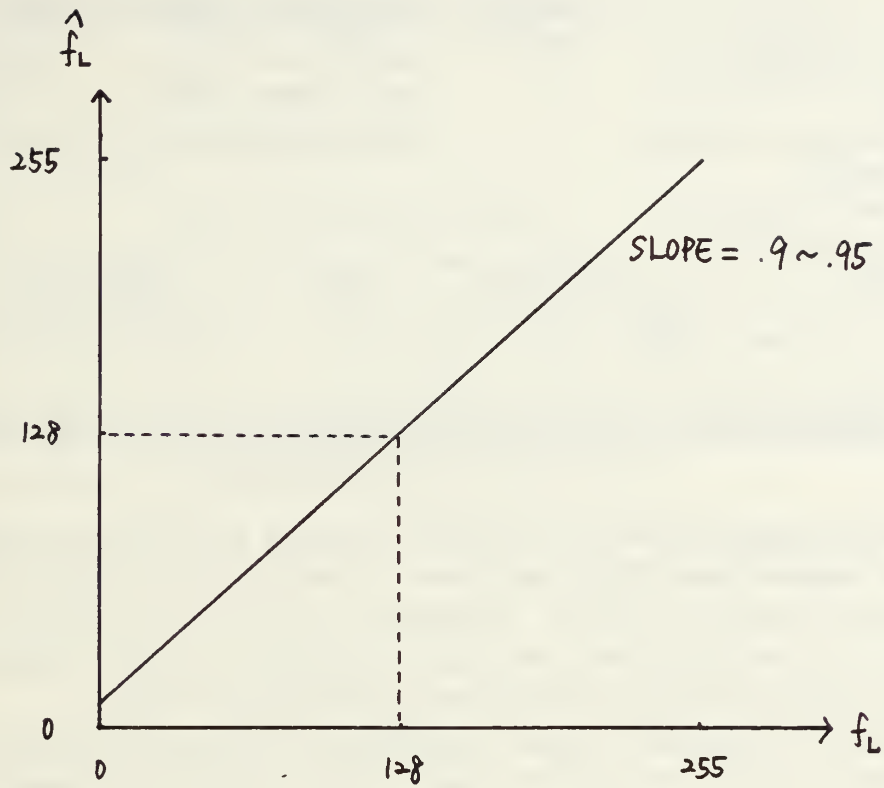

Figure 3. Recommended Choice of Nonlinearity

	Additions	Multiplications	Table Look-ups
Low-pass filtering	M^2N^2	N^2	
Subtraction of $f_L(n_1, n_2)$	N^2		
$k(f_L)$			N^2
Multiplication with $k(f_L)$		N^2	
Nonlinearity			N^2
Addition of $\hat{f}_H(n_1, n_2)$ and $\hat{f}_L(n_1, n_2)$	N^2		
Total	$(M^2+2)N^2$	$2N^2$	$2N^2$

From the above result,

Computations/sec: $(M^2+2)N^2R$ additions, $2N^2R$ multiplications, $2N^2R$ table look-up operations

As an example, when $N = 512, M = 5$, and $R = 30$ (full video rate), we require approximately 200M additions, 15M multiplications, and 15M table look-up operations per second. As another example, when $N = 512, M = 5$, and $R = 1$ (one frame/sec), we require approximately 7M additions, $\frac{1}{2}M$ multiplications and $\frac{1}{2}$M table look-up operations per second.

It may be possible to reduce computations that arise from the low-pass filtering operation, which is the major computational requirement in the algorithm. The low-pass filtered version $f_L(n_1, n_2)$ has a low-pass characteristic and we may be able to undersample it. If we undersample $f_L(n_1, n_2)$ by a factor of $S \times S$, we can reduce the computations required in the low-pass filtering operation by a factor of S^2 at the expense of an increase of approximately N^2 additions and N^2 multiplications per frame required for interpolation of $f_L(n_1, n_2)$. There will be some decrease in performance, but the level of performance decrease is not expected to be too serious. We recommend that future study include a study on this trade-off between performance decrease and reduction in computations.

To determine the storage requirements, we assume that the video data are raster-scanned as in the NTSC signal. We also assume that one memory unit contains one pixel intensity, which is typically represented with 8 bits (1 byte). The storage requirements are as follows:

$f(n_1, n_2)$	:	MN memory units
$f_L(n_1, n_2)$	:	MN memory units
$k(f_L)$	:	256 memory units
Nonlinearity	:	256 memory units
Total	:	$2MN + 512$ memory units

From the above result,

Storage Requirements : $2MN + 512$ memory units

For $N = 512$ and $M = 5$, the algorithm requires about 6K memory units. This memory requirement is not an issue in the real time implementation of the adaptive filtering algorithm.

3 Modified Adaptive Filtering

3.1 Basic Algorithm

To account for the noise visibility problem associated with the adaptive filtering method, one approach proposed and studied by Franco [2] is to enhance the contrast using the adaptive filtering algorithm and then to apply a noise reduction system to reduce the background noise that has been accentuated. Even though this approach is interesting, it has several problems. In a typical noise reduction system, the local luminance mean $f_L(n_1, n_2)$ is often computed and cascading the adaptive filtering method with a noise reduction system requires computation of $f_L(n_1, n_2)$ twice. In addition, a noise reduction system often requires an estimate of the noise variance. The noise variance depends on the contrast enhancement factor $k(f_L)$ used in the adaptive filtering algorithm, and this complicates the design of the noise reduction system.

A simple method which in a sense integrates contrast enhancement with noise reduction has been developed. We'll refer to this method as modified adaptive filtering algorithm. The algorithm is shown in Figure 4. The main difference between this algorithm and the adaptive filtering method discussed in Section 2 is that the contrast enhancement factor k is now a function of the local luminance mean f_L and the local variance σ_f^2. The local variance can be computed approximately from $f_H(n_1, n_2)$ by

$$\sigma_f^2(n_1, n_2) = \frac{1}{L^2} \sum_{k_1=n_1-\frac{L-1}{2}}^{n_1+\frac{L-1}{2}} \sum_{k_2=n_2-\frac{L-1}{2}}^{n_2+\frac{L-1}{2}} f_H^2(k_1, k_2)$$

where L is assumed to be an odd integer and the region over which σ_f^2 is computed has size of $L \times L$ pixels. The normalization factor L^2 can be accounted for in determining $k(f_L, \sigma_f^2)$ and therefore can be ignored.

When σ_f^2 is small, the area is likely to correspond to a region with little signal component (uniform background region) and contrast enhancement in the region is likely to boost only the background noise. When σ_f^2 is very large, the signal component may have large energy and a very large contrast enhancement may not be necessary. When σ_f^2 is in the mid-range, a weak signal component is likely to be present and a large contrast enhancement

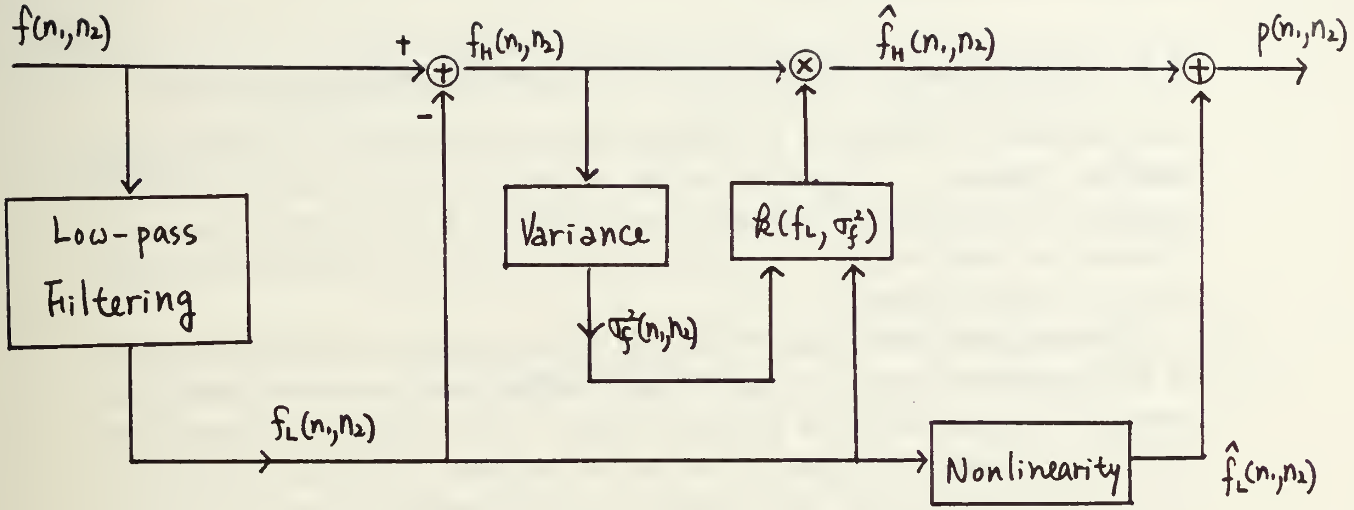

Figure 4. Modified Adaptive Filtering Algorithm

is likely to be beneficial. Even though the noise in this region will be accentuated, the region involved is only a fraction of the image frame and the presence of the accentuated signal component can mask the noise to some extent. The dependence of $k(f_L, \sigma_f^2)$ on σ_f^2 allows us to modify the contrast as a function of σ_f^2.

3.2 Parameter Choice and Expected Performance

The algorithm in Figure 4 is currently being implemented by Lieutenant Roberto Ventura. At the time of this report, we do not have results of applying this method to the data from NUWES. It is expected, however, to solve the noise visibility problem associated with the adaptive filtering method to some extent.

Choosing the parameters of the algorithm requires careful evaluation of the processed images. As an initial starting point, we recommend the parameter choice shown in Figures 5 and 6. The choice of a separable function $k(f_L, \sigma_f^2) = k_1(f_L)k_2(\sigma_f^2)$ is due to the desire to decouple the effects of f_L and σ_f^2 on $k(f_L, \sigma_f^2)$ and reduction in the required storage. If storage is not a problem, $k(f_L, \sigma_f^2)$ can be precomputed and stored. The choice of $k_1(f_L)$ is based on the same considerations as in Section 2.2. The choice of $k_2(\sigma_f^2)$ is chosen based on several considerations. When σ_f^2 is small, there is likely to be little signal component and large contrast enhancement will only result in noise enhancement. When σ_f^2 is very large, the signal component is already very strong and there is not much need for a large contrast increase.

3.3 Computational Requirements

The modified adaptive filtering method in Figure 4 is more expensive computationally than the adaptive filtering method in Figure 1. The additional computations per second are

Computation of Variance: L^2N^2R multiply/adds.
Computation of $k(f_L, \sigma_f^2)$: N^2R multiplications, N^2R table lookups.

Combining this result with the results in Section 2.3,

Computations/sec: $(M^2 + 2)N^2R$ additions, 3 N^2R multiplications,

$$k(f_L, \sigma_f^2) = k_1(f_L) \cdot k_2(\sigma_f^2)$$

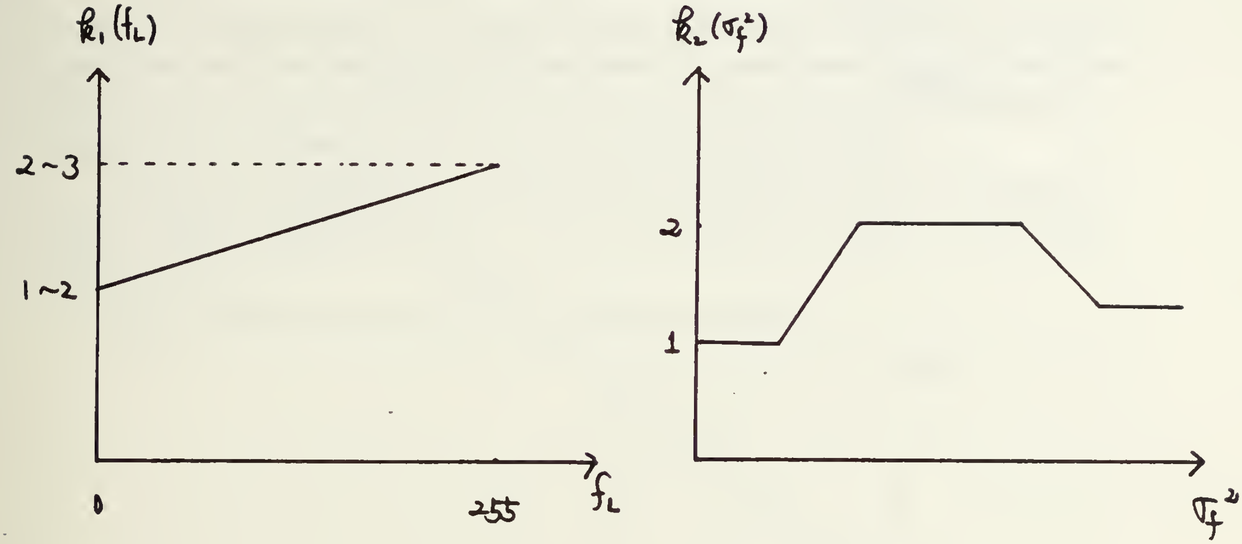

Figure 5. Recommended Initial Choice of $k(f_L, \sigma_f^2)$ for the Modified Adaptive Filtering Algorithm.

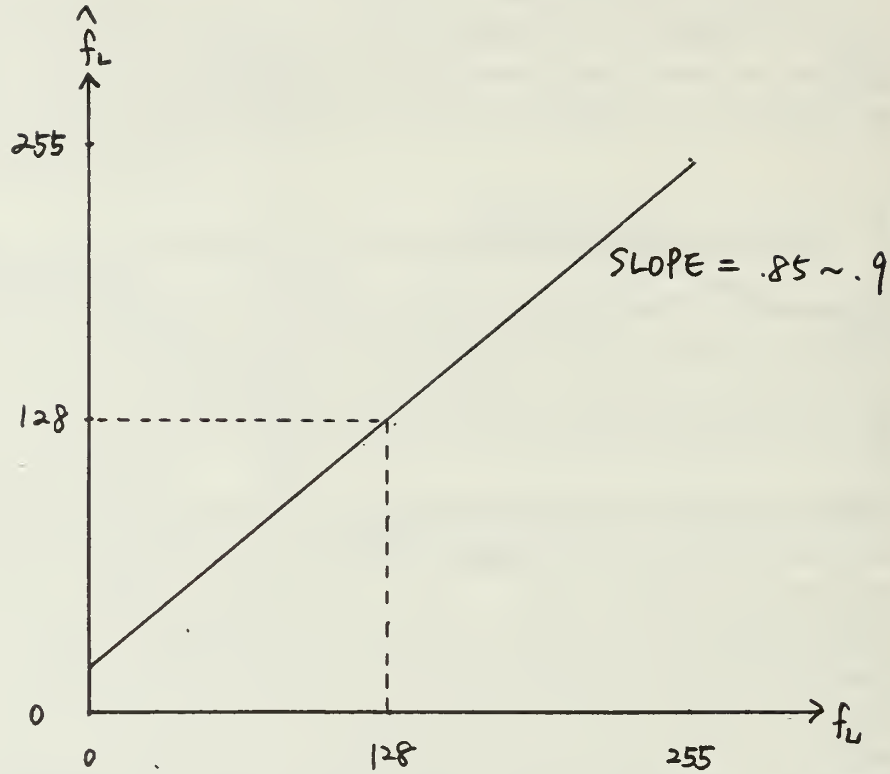

Figure 6. Recommended Initial Choice of Nonlinearity for the Modified Adaptive Filtering Algorithm

<u>L^2N^2R multiply/adds, $3N^2R$ table look-up operations</u>

If we assume that $L = M$ and one multiply/add operation, one add operation, and one multiply operation take the same computation time, the computational requirements of the modified adaptive filtering algorithm is approximately twice that of the adaptive filtering algorithm.

The storage requirements are essentially the same as that of the adaptive method. The small increase is due to the storage of $k_2(\sigma_f^2)$. The storage requirements are

<u>Storage Requirements: $2MN + 768$ memory units</u>

4 Temporal Filtering

4.1 Basic Algorithm

Careful observation of the video data from NUWES shows that some form of frame averaging may result in image enhancement. Specifically, in a typical torpedo recovery operation, the underwater camera is often stationary or moves slowly. As a result, objects of interest such as a torpedo displayed on the monitor do not appear to change much from one frame to the next frame. However, the elements such as water bubbles and stirred- up sediment that degrade the visibility of objects of interest appear to change quite rapidly. Some form of frame averaging, therefore, has potential to add the signal component constructively while adding signals from the degrading sources destructively. This can result in improvement of the signal to noise ratio (SNR).

Frame averaging can involve storage of the frames involved, which can increase the storage requirements drastically. One method that performs frame averaging without requiring storage of more than one frame at a time is first order recursive temporal filtering. Consider one particular pixel. Let $f(n)$ denote the intensity of the image at the nth frame at that particular pixel. Note that the variable "n" is a time variable, not a spatial variable. The processed image $p(n)$ in the first order recursive temporal filtering is given by

$$p(n) = (1-a) \cdot p(n-1) + a \cdot f(n),$$

where "a" is a constant between 0 and 1. As $f(n)$ is received and once $p(n)$ is computed, $p(n-1)$ is no longer needed and can be replaced with $p(n)$. At any given time, therefore, we need to store only one frame of data. The system function $H(z)$ and unit sample response of the system $h(n)$ are given by

$$H(z) = \frac{a}{1-(1-a)z^{-1}} \quad ,$$

and

$$h(n) = a(1-a)^n u(n).$$

As "a" approaches 0, many frames are averaged. When $a = 1$, no temporal filtering is performed.

Many variations of the above recursive filtering method are possible. For example, by initializing $p(n)$ once in a while, choosing "a" close to zero, and normalizing the result, a result essentially the same as simple FIR filtering can be obtained. We also note that the contrast enhancement methods discussed in Sections 2 and 3 can also be applied to the temporally filtered images, if desired.

4.2 Parameter Choice and Expected Performance

A sequence of image frames that have been digitized are not available and we were not able to process the data using the temporal filtering algorithm discussed above. To the extent that our assumption that objects of interest do not change rapidly between consecutive frames while the degrading sources do is valid, the algorithm is expected to perform very well.

To see the expected performance improvement in an ideal environment, suppose $f(n)$ can be expressed as

$$f(n) = s(n) + w(n)$$

where $s(n)$ is the signal which is constant independent of n and $w(n)$ is zero-mean white noise with variance of σ_w^2. The processed image $p(n)$ can be expressed as

$$p(n) = s(n) + w_p(n)$$

where $w_p(n)$ is zero-mean white (in the spatial domain) noise with variance of $\frac{a^2}{1-(1-a)^2} \cdot \sigma_w^2$. When $a = 0.2$, the noise variance reduction is by approximately a factor of 10, corresponding to 10dB SNR improvement.

In practice, of course, the assumptions made in the above analysis will not be valid. The signal, for example, will change as a function of time and this will cause signal blurring. There is potential for significant image enhancement, however, and we recommend that future studies include the application of temporal filtering to the data from NUWES. Our initial recommendation for the choice of "a" is 0.2.

4.3 Computational Requirements.

The computations involved are N^2 additions and $2N^2$ multiplications per frame. Therefore, Computations/sec: N^2R additions, $2N^2R$ multiplications

When $N = 512$ and $R = 30$, the computational requirements are $7\frac{1}{2}M$ additions and 15M multiplications per second.

The storage requirement is the storage of one image frame and therefore,

Storage requirement: N^2 memory units

5 Pseudo Color Representation

It is well known that the human visual system is quite sensitive to color. The number of distinguishable intensities, for example, is much less than the number of distinguishable colors and intensities. As a result, when a black and white image is displayed using color, the result may not be a natural looking image but the contrast of the image may be considerably improved. In addition, color images are generally much more pleasant to look at than black and white images.

Mapping a black and white image $f(n_1, n_2)$ to a color image involves generation of the red component $f_R(n_1, n_2)$, the green component $f_G(n_1, n_2)$ and the blue component $f_B(n_1, n_2)$. Designing the specific transformation that maps $f(n_1, n_2)$ to $f_R(n_1, n_2)$, $f_G(n_1, n_2)$ and $f_B(n_1, n_2)$ is limited only by one's artistic imagination, and involves a fair amount of trial and error. One recommendation we have for the design of the transformation table is that similar colors be used for similar pixel intensities. Otherwise, the resulting color image will appear noisy. Another recommendation is that low intensities be mapped to blue and high intensities be mapped to red. The intensities in between can perhaps be mapped following the rainbow colors. It is well known that people perceive blue as "cold" or "dark" and perceive red as "hot" or "bright".

Once the transformation table is designed, processing involves only N^2R table look-up operations per second and storage requirement for the table is minimal. Allowing table look-up operations prior to image display is a common feature in commercially available real time video data digitization and display systems. Even though the level of performance improvement is not clear, it is a worthwhile approach to be explored in future study. Pseudo-color operation, of course can be performed in addition to other methods discussed in previous sections.

6 Hardware Implementation

To apply the algorithms discussed in previous sections to enhance the efficiency of the torpedo recovery operation, their real time implementation is essential. In this section we discuss some very preliminary studies we performed to determine the feasibility of implementing the algorithms in real time at a reasonable cost.

One cost-effective approach to real time implementation is to interface a signal processing (SP) chip to a personal computer (PC) which has been interfaced to a unit such as the PC Vision board that is capable of digitizing and displaying the video data in real time, as shown in Figure 7. Using a PC as an overall control unit and interfacing an image digitization and display unit such as the PC Vision board is a very cost-effective way to acquire and display video data. The algorithms we wish to implement in real time require a very large number of arithmetic operations and the SP chip is necessary to perform the arithmetic operations very fast. The adaptive filtering method discussed in Section 2 has already been implemented [3] using the approach in Figure 7. With a NEC μPD77230 chip and PC-9800, processing one image frame using the adaptive filtering method discussed in Section 2 with N=256 and M=17 requires approximately 3 seconds.

In studying the feasibility of real time implementation, a number of issues have to be considered. In this very preliminary study, however, we have made a variety of simplifying assumptions. Specifically, we assume that storage requirement is not an issue in real time implementation. Since the storage requirement for data is in the order of N^2 memory units or less, this assumption is reasonable. We have also assumed that the computational time required for each of an add, a multiply, and a multiply/add operation is the same. As floating point arithmetic operations are becoming more common, this assumption also appears reasonable. We'll refer to an addition, a multiplication, or a multiply/add operation as one arithmetic operation. We denote the computation time for one arithmetic operation by c sec. For currently available SP chips such as Texas Instrument's most recent TMS320 family chip, the computation time "c" is in the order of 100 nanosec (10^{-9} sec). We also assume that the computation time required for all arithmetic operations is a fraction (p%) of the total amount of time required for a given algorithm. The fraction "p" % depends on a number

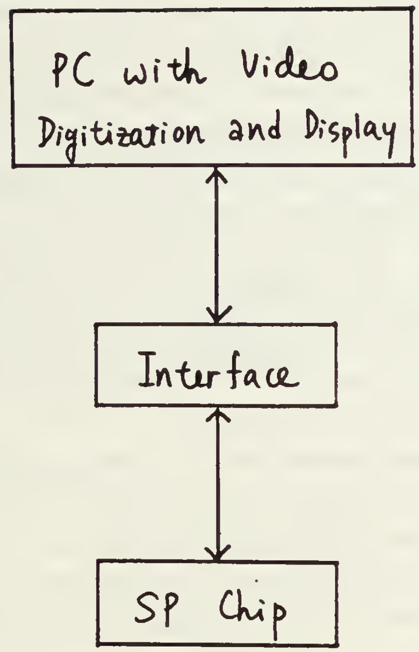

Figure 7. One Cost-effective Approach to Real Time Implementation of Image Enhancement Algorithms

of factors such as data transfer time, algorithm complexity, the specific SP chip, PC, and interface used. We'll consider two different reasonable values of p in our analysis.

Under the assumptions and notations discussed above, we have computed the amount of time required for processing 1 second of video data for various choices of the parameters. The results are shown in Table 1. If the required processing time is less than 1 second, it implies that real time implementation is possible. For the specific choices of the parameters used in the table, none of the cases can be implemented in real time. However, some of the cases such as temporal filtering are not far from real time. There are a variety of ways to reduce computational requirements. For example, as we discussed in Section 2, we may be able to reduce computations required for the low-pass filtering operation by under-sampling $f_L(n_1, n_2)$. It is also possible to increase the computational speed. For example, the computation time required for an arithmetic operation is becoming smaller as new SP chips are introduced. In addition, the algorithms have very simple structures and we can process the data in parallel using more than one SP chip.

In summary, our preliminary study indicates that real time implementation of the algorithms discussed in previous sections at a reasonable cost is not a simple task due to a very large amount of data involved. However, it appears to be within a reachable goal with the hardware technology that is currently available or will shortly be available.

TABLE I

Algorithms	Choice of c and p			
—	$c = 100$ nsec $p = 25\%$	$c = 100$ns $p = 50\%$	$c = 200$ nsec $p = 25\%$	$c = 100$ nsec $p = 50\%$
Adaptive Filtering $N = 512, M = 5, R = 30$	90 sec	45 sec	180 sec	90 sec
Adaptive Filtering $N = 512, M = 5, R = 1$	3 sec	1.5 sec	6 sec	3 sec
Modified Adaptive Filtering $N = 512, M = 5, L = 5, R = 30$	180 sec	90 sec	360 sec	180 sec
Modified Adaptive Filtering $N = 512, M = 5, L = 5, R = 1$	6 sec	3 sec	12 sec	6 sec
Temporal Filtering $N = 512, R = 30$	9 sec	4.5 sec	18 sec	9 sec

7 References

1. T. Peli and J. S. Lim, "Adaptive Filtering for Image Enhancement," J. of Optical Engineering, vol. 21, pp. 108–112, 1982.

2. J. A. Franco, "Enhancement of Video Images Degraded by Turbid Water," Master's Thesis, Department of Elec. and Comp. Eng., Naval Postgraduate School, Monterey, Cal., Dec. 1986.

3. T. Nishitani, I. Kuroda, S. Matsukawa, J. Sunaga, and H. Koyama, "A Single-chip 32 Bit Floating-Point Signal processor and It's Application to Image Data Processing."

INITIAL DISTRIBUTION LIST

Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
Department Chairman, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943	1
Mr. Richard Evans, Code 7021 NUWES Code 7021 Naval Undersea Weapons Engineering Station Keyport, Washington 98345	2
Ms. Tami Peli Gr. 21 M. I. T. Lincoln Laboratory Lexington, Massachusetts 02173-0073	1
Professor Jae Lim Department of Electrical and Computer Engineering Room 36-653 M. I. T. Cambridge, Massachusetts 02139	1
Professor O. B. Wilson, Code 61Wl Department of Physics Naval Postgraduate School Monterey, California 93943	1
Director of Research Administration Code 012 Monterey, CA 93943	1
Mr. Robert Marimon Code 70 Naval Undersea Warfare Eng. Station Keyport, WA 98345	1
CDR Hiller Code 80 Naval Undersea Warfare Eng. Station Keyport, WA 98345	1

Dr. L. Stotts
Bldg. 33 Room 3037
NOSC
San Diego, CA 92152-5000 1